



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

mw

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/782,137	02/19/2004	Paul R.M. Carpentier	HYPRP002	5213
22434	7590	08/15/2006	EXAMINER	
BEYER WEAVER & THOMAS, LLP			LU, KUEN S	
P.O. BOX 70250			ART UNIT	
OAKLAND, CA 94612-0250			PAPER NUMBER	
			2167	

DATE MAILED: 08/15/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.		Applicant(s)	
	10/782,137		CARPENTIER ET AL.	
	Examiner		Art Unit	
	Kuen S. Lu		2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 19 February 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-29 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-29 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 07 July 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>3/14/2005</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. The Action is responsive to Applicant's Application filed February 19, 2004. Claims 1-29 are pending.

Priority

2. Applicant's claim for the benefit of a prior-filed application, U.S. provisional patent application No. 60/449,172, filed February 21, 2003, under 35 U.S.C. §119(e) or under 35 U.S.C. 120, 121, or 365(c) is acknowledged.

Since the provisional application relied upon as a priority document contains a less detailed disclosure of the invention, the claim of priority will be considered on a claim-by-claim basis. The priority date of the instant application is at latest 14 February 2004 (the filing date), but depending upon the specific material claimed, could be as early as February 21, 2003.

Information Disclosure Statement

3. The information disclosure statements submitted March 14, 2005 was filed before the mailing date of the first office action. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered and corresponding PTO-1449 is electronically signed and attached. Also acknowledged is Applicant's incorporation of International Publication Numbers WO 99/38093 dated 7/29/1999, WO 99/38092 dated 7/29/1999 and WO 01/18633 dated 3/15/2001 as reference.

Drawings

4. The drawings, filed July 7, 2004, in which Figs. 1-12 and 14-16B are considered in compliance with 37 CFR 1.81 and accepted. Fig. 13 is objected to because of reason described below.

3.1. Fig. 13 is objected to, under 37 CFR 1.83(a), because it fails to show two decision steps as described in Specification. In Fig. 13, element 558 is a processing block for computing a hash value, however, the block results in two outputs and it seems missing a decision step for determining CHECK or NO CHECKS. Further, a YES/NO decision step at element 562 generates third output, other than YES and NO. It seems a further step, following YES decision, is missing for determining next processing elements, block 572 or 584. Any structural detail that is essential for a proper understanding of the disclosed invention should be shown in the drawing. MPEP § 608.02(d). Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either

"Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5.1. Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hunt et al. (U.S. Patent Application 2003/0005306, hereafter "Hunt") in view of Masinter (U.S. Patent 5,742,807).

As per claim 1, Hunt teaches "A method of adding a computer file to a database" (See Fig. 5 and Page 3, [0031] where file is copied to repository), said method comprising:
"receiving said computer file to be added to said database" (See Fig. 3 and Page 3, [0031] where a current file is to be copied to repository);

“computing a first hash value for said file using a first hash function” (See Fig. 5 and Page 3, [0027] where a message digest generated for a current file based on file content when a first cache operation was performed); and

“computing a second hash value for said file using a second hash function” (See Fig. 6 and Page 3,[0032] where a new cryptographic hashes based on content of message digests stored at the client and corresponding entry in the database of message digests at the repository are generated for file synchronization verification processing).

Hunt does not explicitly teach “storing said file in said database at a location identified by said first hash value”, although Hunt teaches uniquely identifying a file stored in the client.

However, Masinter teaches “storing said file in said database at a location identified by said first hash value” (See Fig. 2 and col. 4, lines 40-54 where entry in the hash-to-location index is the unique identifier for mapping and identifying hash to document storage location).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Masinter with Hunt reference by implementing hash-to-location index for integrating file storage location with hash because both references are directed to hashing document for storage where Masinter teaches balancing interests between open repository and access control while Hunt teaches archiving only files changed, added or deleted files for avoiding unnecessary copying, and the combined teaching would have enabled Hunt's system to improve access control and, at the same time, maintain system integrity because of the scheme

of mapping document index with storage location and hash. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Masinter and Hunt references further teaches the following:

“adding said first hash value and said second hash value to a data structure associated with said database, wherein said first and second hash values are associated with one another in said data structure, whereby said addition of said first and second hash values to said data structure indicates the presence of said file in said database” (See Hunt: Figs. 5-6 and Page 3, [0031]-[0032] where message digest for the file to be stored and a cryptographic hash for message digests are stored in a repository database where cryptographic hash is generated based on content of message digest and message digest is generated based on file content).

As per claim 2, the combined teaching of Masinter and Hunt references further teaches the following:

“determining whether a copy of said file is present in said database” (See Hunt: Fig. 5 and Page 3, [0031] where a decision was made to determine if a current file to be copied to repository);

“when it is determined that a copy of said file is present in said database, computing a verification hash value for said copy of said file using said second hash function and comparing said second hash value to said verification hash value” (See Fig. 6 and Page

3, [0027] and [0032] where message digests are generated for comparing files and cryptographic hashes are generated for comparing message digests) and “taking an action when it is determined that said second hash value matches said verification hash value” (See Fig. 6 and Page 3, [0032] where a cryptographic hash is compared with message digests and skip synchronization process if the comparison is matched).

As per claim 3, the combined teaching of Masinter and Hunt references further teaches “determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said first hash value” (See Masinter: Fig. 7, elements 100-104 and col. 6, lines 34-40 where hash generated for a file is searched in the hash-to-location index, and Hunt: Fig. 2, element 230 and Page , [0028] where message digest is stored in repository).

As per claim 4, the combined teaching of Masinter and Hunt references further teaches “determining whether a copy of said file is present in said database by searching said data structure for a value that matches said first hash value” (See Masinter: Fig. 7, elements 100-104 and col. 6, lines 34-40 where hash generated for a file is searched in the hash-to-location index, and Hunt: Fig. 2, element 230 and Page 3, [0028] where message digest generated at the client is compared with message digest stored in repository).

As per claim 5, the combined teaching of Masinter and Hunt references further teaches “data structure is a table, a list or a tree data structure” (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

As per claim 6, the combined teaching of Masinter and Hunt references further teaches “second hash function provides stronger security than said first hash function” (See Hunt: Fig. 5 and Page 3, [0027] and [0031]-[0032] where message digest is the first hash generated based on file content and second cryptographic hash is second hash generated based on message digest, the first hash, suggests having stronger security).

As per claim 7, Hunt teaches “A method of retrieving a desired computer file from a database” (See Fig. 2 and Page 3, [0026]-[0027] where files are stored and retrieved from repository), said method comprising:
“obtaining a unique identifier for said desired computer file” (See Fig. 2 and Page 3, [0026]-[0027] where message digest is a unique identifier for the file stored in client and to be retrieved for being cached in repository).

Hunt does not explicitly teach “retrieving a stored file from said database using said unique identifier as a reference”.

However, Masinter teaches “retrieving a stored file from said database using said unique identifier as a reference” (See Fig. 2 and col. 4, lines 40-54 where entry in the

hash-to-location index is the unique identifier for mapping and identifying hash to document storage location).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Masinter with Hunt reference by implementing hash-to-location index for integrating file storage location with hash because both references are directed to hashing document for storage where Masinter teaches balancing interests between open repository and access control while Hunt teaches archiving only files changed, added or deleted files for avoiding unnecessary copying, and the combined teaching would have enabled Hunt's system to improve access control and, at the same time, maintain system integrity because of the scheme of mapping document index with storage location and hash. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Masinter and Hunt references further teaches the following:

"retrieving a first hash value-second hash value pair from a data structure associated with said database by using said unique identifier" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where message digest and cryptographic hash were retrieved from repository for comparison with the digest and hash at the client, respectively, and Masinter: Fig. 2 and col. 4, lines 40-54 where entry in the hash-to-location index is the unique identifier for mapping and identifying hash to document storage location), "said unique identifier matching said first hash value, wherein said first hash value has been derived from said stored file using a first hash function and wherein

said second hash value has been derived from said stored file using a second hash function" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where the message digest is derived from file contents and cryptographic hash is derived from message digests of files);

"computing a verification hash value for said stored file using said second hash function" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests);

"comparing said verification hash value to said second hash value" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash at the client is compared with that on the repository) and

"determining that said stored file is said desired file when said verification hash value matches said second hash value, whereby said desired file is retrieved from said database" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash at the client is compared with that on the repository and determine if synchronization process to be perform where files are retrieved for copying from client to repository).

As per claim 8, the combined teaching of Masinter and Hunt references further teaches the following:

"computing an addressing hash value for said stored file using said first hash function" (See Masinter: Fig. 2 and col. 4, lines 40-54 where entry in the hash-to-location index is the unique identifier for mapping and identifying hash to document storage location);

“comparing said addressing hash value to said first hash value” (See Masinter: Fig. 3 and col. 4, line 63 – col. 5, line 2 where hash is looked up in the hash-to-location index for document retrieval) and

“indicating that said second hash value matches said addressing hash value” (See Masinter: Fig. 3 and col. 4, line 63 – col. 5, line 2 where hash is looked up in the hash-to-location index for document retrieval).

As per claim 9, the combined teaching of Masinter and Hunt references further teaches “when it is determined that said stored file is said desired file, delivering said stored file to a user” (See Masinter: Fig. 3 and col. 4, line 63 – col. 5, line 2 where hash is looked up in the hash-to-location index for document retrieval).

As per claim 10, the combined teaching of Masinter and Hunt references further teaches “performing said step of retrieving a stored file by searching said data structure for said unique identifier” (See Masinter: Fig. 3 and col. 4, line 63 – col. 5, line 2 where hash is looked up in the hash-to-location index for document retrieval).

As per claim 11, the combined teaching of Masinter and Hunt references further teaches “data structure is a table, a list or a tree data structure” (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

As per claim 12, the combined teaching of Masinter and Hunt references further teaches "second hash function provides stronger security than said first hash function" (See Hunt: Fig. 5 and Page 3, [0027] and [0031]-[0032] where message digest is the first hash generated based on file content and second cryptographic hash is second hash generated based on message digest, the first hash, suggests having stronger security).

As per claim 13, Hunt teaches "A method of adding hash authority functionality to a database of files" (See Fig. 2 and Page 3, [0026]-[0027] where message digests are generated based on file content for files in client), said method comprising: "creating a hash authority data structure said data structure including a plurality of entries" (See Fig. 2 and Page 3, [0026]-[0027] where message digests are stored in database table).

Hunt does not explicitly teach "retrieving an addressing hash value for a first file of said database said addressing hash value having been computed from said first file using an addressing hash function".

However, Masinter teaches "retrieving an addressing hash value for a first file of said database said addressing hash value having been computed from said first file using an addressing hash function" (See Fig. 2 and col. 4, lines 40-54 where entry in the hash-to-location index is the unique identifier for mapping and identifying hash to document storage location).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Masinter with Hunt reference by implementing hash-to-location index for integrating file storage location with hash because both references are directed to hashing document for storage where Masinter teaches balancing interests between open repository and access control while Hunt teaches archiving only files changed, added or deleted files for avoiding unnecessary copying, and the combined teaching would have enabled Hunt's system to improve access control and, at the same time, maintain system integrity because of the scheme of mapping document index with storage location and hash. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Masinter and Hunt references further teaches the following:

"retrieving said first file from said database" (See Masinter: Fig. 2 and col. 4, lines 40-54 where entry in the hash-to-location index is the unique identifier for mapping and identifying hash to document storage location);

"computing a verification hash value for said first file using a verification hash function" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash at the client is compared with that on the repository);

"adding said addressing hash value and said verification hash value to one of said entries of said hash authority data structure" (See Masinter: Fig. 2 and col. 4, lines 40-54 where entries of hash-to-location are stored in an index, and Hunt: Fig. 5 and Page

3, [0027] and [0031]-[0032] where message digest and cryptographic hash are stored in database tables),

“said addressing hash value and said verification hash value being associated with one another in said database” (See Masinter: Fig. 2 and col. 4, lines 40-54 where hash and file storage location are associated as entry to the hash-to-location index).

As per claim 14, the combined teaching of Masinter and Hunt references further teaches “hash authority data structure is a table, a list or a tree data structure” (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

As per claim 15, the combined teaching of Masinter and Hunt references further teaches “verification hash function provides stronger security than said addressing hash function” (See Hunt: Fig. 5 and Page 3, [0027] and [0031]-[0032] where message digest is the first hash generated based on file content and second cryptographic hash is second hash generated based on message digest, the first hash, suggests having stronger security).

As per claim 16, the combined teaching of Masinter and Hunt references further teaches “performing the final four steps of claim 13 for each of the other of said plurality of files in said database” (See Masinter: Fig. 2 and col. 4, lines 40-54 where entry in the hash-to-location index is the unique identifier for mapping and identifying hash to

document storage location; Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash at the client is compared with that on the repository; Masinter: Fig. 2 and col. 4, lines 40-54 where entries of hash-to-location are stored in an index, and Hunt: Fig. 5 and Page 3, [0027] and [0031]-[0032] where message digest and cryptographic hash are stored in database tables; and Masinter: Fig. 2 and col. 4, lines 40-54 where hash and file storage location are associated as entry to the hash-to-location index).

As per claim 17, Hunt teaches "A method of upgrading a verification hash function in a database of files" (See Fig. 6 and Page 3, [0032] where verification cryptographic hash is updated when the hash is generated based on message digests in the verification process), said method comprising:

"accessing a data structure representing said database of files" (See Fig. 2 and Page 3, [0026]-[0027] where tables storing message digest, generated based on file content, and cryptographic hash are the data structure representing files).

Hunt does not explicitly teach "said data structure including an addressing hash value-verification hash value pair for each of said files in said database, wherein said addressing hash values have been computed using an addressing hash function, and wherein said verification hash values have been computed using said verification hash function", although Hunt teaches generating message digest and verification cryptographic hash for files and digests, respectively, as described above.

However, Masinter teaches "said data structure including an addressing hash value-verification hash value pair for each of said files in said database, wherein said addressing hash values have been computed using an addressing hash function, and wherein said verification hash values have been computed using said verification hash function" (See Fig. 2 and col. 4, lines 40-54 where hash for file is generated and an entry consisting of hash and file storage location is stored in the hash-to-location index as an unique identifier for mapping and identifying hash to document storage location).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Masinter with Hunt reference by implementing hash-to-location index for integrating file storage location with hash because both references are directed to hashing document for storage where Masinter teaches balancing interests between open repository and access control while Hunt teaches archiving only files changed, added or deleted files for avoiding unnecessary copying, and the combined teaching would have enabled Hunt's system to improve access control and, at the same time, maintain system integrity because of the scheme of mapping document index with storage location and hash. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Masinter and Hunt references further teaches the following:

"for each of said files in said database, retrieving the file, computing a new verification hash value for the file using a stronger verification hash function, and adding said new verification hash value to said data structure in association with said addressing hash

value of said file" (See Hunt: Fig. 5 and Page 3, [0027] and [0031]-[0032] where message digest is the hash generated based on file content for each file and the verification cryptographic hash is the hash generated based on message digest suggesting having stronger security, and Masinter: Fig. 3 and col. 4, line 63 – col. 5, line 2 where hash is looked up in the hash-to-location index for document retrieval).

As per claim 18, the combined teaching of Masinter and Hunt references further teaches "A method as recited in claim 17 wherein said data structure is a table, a list or a tree data structure" (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

5.2. Claims 19-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hunt et al. (U.S. Patent Application 2003/0005306, hereafter "Hunt") in view of Oom Temudo de Castro et al. (U.S. Patent Application 2004/0111608, hereafter "Castro").

As per claim 19, Hunt teaches "A method of adding a computer file to a database ..." (See Fig. 2 and Page 3, [0026] where files are to be copied to repository), said method comprising:

"receiving said computer file to be added to said database" (See Fig. 2 and Page 3, [0026] where files are to be copied to repository).

Hunt does not explicitly teach that the computer file being added to database by using a random number, although Hunt teaches using hash to secure file distribution to the repository at Fig. 6 and Page 3, [0032].

However, Castro teaches distributing file to a storage location by a combination of factors, including cryptographic random number (See Page 3, [0036]).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Castro with Hunt reference by combining cryptographic random number with hash value to distribute files to the repository because both references are directed to distributing files to repository storage where effective operation of distributing files and secure protection of data are both critical, and the combined teaching of the operation would have enabled Hunt's system to improve access control and, at the same time, maintain data integrity because the file distribution scheme would have combined random number and hash such that location of files would have been transparent to users and the random number would have been cryptographically generated. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Castro and Hunt references further teaches the following: "generating a random number for said file using said random number generator" (See Castro: Page 3, [0036] where a random number self suggests generated by a random number generator);

“computing a verification hash value for said file using a verification hash function” (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests);

“storing said file in said database at a location identified by said random number” (See Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number) and

“adding said random number and said verification hash value to an entry in a data structure associated with said database, wherein said random number and said verification hash value are associated with one another in said data structure, whereby the addition of said entry in said data structure indicates the presence of said file in said database” (See Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number, and Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests and at Fig. 5 and Page 3, [0031] where file is copied to repository if corresponding message digest does not match).

As per claim 20, the combined teaching of Castro and Hunt references further teaches “determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said verification hash value” (See Hunt: Fig. 5 and Page 3, [0031] where file is copied to repository if corresponding message digest does not match).

As per claim 21, the combined teaching of Castro and Hunt references further teaches “wherein said data structure is a table, a list or a tree data structure” (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

As per claim 22, Hunt teaches “A method of adding a computer file to a database ...” (See Fig. 2 and Page 3, [0026] where files are to be copied to repository), said method comprising:

“receiving said computer file to be added to said database” (See Fig. 2 and Page 3, [0026] where files are to be copied to repository).

Hunt does not explicitly teach that the computer file being added to database by using a random number, although Hunt teaches using hash to secure file distribution to the repository at Fig. 6 and Page 3, [0032].

However, Castro teaches distributing file to a storage location by a combination of factors, including cryptographic random number (See Page 3, [0036]).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Castro with Hunt reference by combining cryptographic random number with hash value to distribute files to the repository because both references are directed to distributing files to repository storage where effective operation of distributing files and secure protection of data are both critical, and the combined teaching of the operation would have enabled Hunt's system to improve access control and, at the same time, maintain data integrity because the file

distribution scheme would have combined random number and hash such that location of files would have been transparent to users and the random number would have been cryptographically generated. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Castro and Hunt references further teaches the following: "accessing a data structure representing said database, said data structure including a random number-verification hash value pair for each of said files in said database, wherein said random numbers have been computed using said random number generator, and wherein said verification hash values have been computed using a verification hash function" (See Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number, and Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests and at Fig. 5 and Page 3, [0031] where file is copied to repository if corresponding message digest does not match); "computing a new verification hash value for said file using said verification hash function" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests each time a verification process is performed); "determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said new verification hash value" (See Hunt: Fig. 5 and Page 3, [0031] where file is copied to repository if corresponding message digest does not match);

"when it is determined that a copy of said file already exists in said database, returning a random number associated with said copy of said file to a user" (See Hunt: Fig. 6 and Page 3, [0032] where a cryptographic hash is generated and compared with message digests and skip synchronization process if the matched hash is maintained), "said random number being associated with said new verification hash value" (See Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number, and Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests).

As per claim 23, the combined teaching of Masinter and Hunt references further teaches "wherein said data structure is a table, a list or a tree data structure" (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

As per claim 24, Hunt teaches "A method of adding a computer file to a database ..." (See Fig. 2 and Page 3, [0026] where files are to be copied to repository), said method comprising:
"receiving said computer file to be added to said database" (See Fig. 2 and Page 3, [0026] where files are to be copied to repository).

Hunt does not explicitly teach that the computer file being added to database by using a random number generator, although Hunt teaches using hash to secure file distribution to the repository at Fig. 6 and Page 3, [0032].

However, Castro teaches distributing file to a storage location by a combination of factors, including cryptographic random number (See Page 3, [0036]).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Castro with Hunt reference by combining cryptographic random number with hash value to distribute files to the repository because both references are directed to distributing files to repository storage where effective operation of distributing files and secure protection of data are both critical, and the combined teaching of the operation would have enabled Hunt's system to improve access control and, at the same time, maintain data integrity because the file distribution scheme would have combined random number and hash such that location of files would have been transparent to users and the random number would have been cryptographically generated. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Castro and Hunt references further teaches the following: "generating a random number for said file using said random number generator" (See Castro: Page 3, [0036] where a random number self suggests generated by a random number generator);

"accessing a data structure representing said database" (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are generated by file content and stored in a table of a database repository),

"said data structure including a random number-verification hash value pair for each of said files in said database, wherein said random numbers have been computed using said random number generator, and wherein said verification hash values have been computed using a verification hash function" (See Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number, and Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests);

"computing a new verification hash value for said file using said verification hash function" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests);

"determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said new verification hash value" (See Hunt: Fig. 5 and Page 3, [0031] where file is copied to repository if corresponding message digest does not match) and

"when it is determined that a copy of said file already exists in said database, adding said first random number and said new verification hash value as an entry in said data structure, retrieving a second random number from said data structure that is associated with said copy of said file, adding a mapping to said entry that maps said first random number to said second random number, whereby a reference to said first

random number is mapped to said second random number to facilitate access to said file" (See Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number, and Hunt: Figs. 2, 5-6, and Page 3, [0027], [0031]-[0032] where message digest is calculated based on file content and cryptographic hash is calculated from message digests, and further at Fig. 5 and Page 3, [0027] and [0031]-[0032] where message digest and cryptographic hash are stored in database tables).

As per claim 25, the combined teaching of Castro and Hunt references further teaches "data structure is a table, a list or a tree data structure" (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

As per claim 26, Hunt teaches "A method of retrieving a desired computer file from a database ..." (See Fig. 2 and Page 3, [0026] where files are retrieved from client for copying to repository).

Hunt does not explicitly teach that the computer file being retrieved by using a random number generator, although Hunt teaches using hash to secure file distribution to the repository at Fig. 6 and Page 3, [0032].

However, Castro teaches distributing file to a storage location by a combination of factors, including cryptographic random number (See Page 3, [0036]).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Castro with Hunt reference by combining cryptographic random number with hash value to distribute files to the repository because both references are directed to distributing files to repository storage where effective operation of distributing files and secure protection of data are both critical, and the combined teaching of the operation would have enabled Hunt's system to improve access control and, at the same time, maintain data integrity because the file distribution scheme would have combined random number and hash such that location of files would have been transparent to users and the random number would have been cryptographically generated. (See BACKGROUND OF THE INVENTION of the references).

The combined teaching of Castro and Hunt references further teaches the following:

- "obtaining a random number associated with said desired computer file" (See Castro: Page 3, [0036] where files are distributed to a storage location by using a combination of factors, including cryptographic random number);
- "said random number having been generated for said file using said random number generator" (See Castro: Page 3, [0036] where a random number self suggests generated by a random number generator);
- "retrieving a stored file from said database using said random number as a reference" (See Castro: Page 3, [0036] where files are distributed to and retrieved from a storage locations based on a combined set of factors, including cryptographic random number);
- "using said random number to lookup an associated first hash value from a data

structure associated with said database, wherein said first hash value has been computed from said stored file using a verification hash function" (See Castro: Page 3, [0036] where files are distributed to and retrieved from a storage locations based on a combined set of factors, including cryptographic random number, and Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests);

"computing a second hash value for said stored file using said verification hash function" (See Hunt: Figs. 5-6, and Page 3, [0031]-[0032] where cryptographic hash is calculated from message digests);

"comparing said first hash value to said second hash value" (See Fig. 6 and Page 3, [0032] where cryptographic hash values are compared in the file verification process) and

"determining that said stored file is said desired file when said first hash value matches said second hash value, whereby said desired file is retrieved from said database" (See Hunt: Fig. 5 and Page 3, [0027] and [0031]-[0032] where message digest and cryptographic hash are stored in database tables where file is copied to repository if corresponding message digest does not match and Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number, and Hunt: Figs. 2, 5-6, and Page 3, [0027], [0031]-[0032] where message digest is calculated based on file content and cryptographic hash is calculated from message digests).

As per claim 27, the combined teaching of Castro and Hunt references further teaches "when it is determined that said stored file is said desired file, delivering said stored file to a user" (See Hunt: Fig. 5 and Page 3, [0027] where message digests are compared to retrieve file for copying to repository).

As per claim 28, the combined teaching of Castro and Hunt references further teaches "performing said step of retrieving a stored file by searching said data structure for said random number" (See Castro: Page 3, [0036] where files are distributed to a storage locations based on a combined set of factors, including cryptographic random number, Hunt: Fig. 5 and Page 3, [0027] where message digests are compared to retrieve file for copying to repository).

As per claim 29, the combined teaching of Castro and Hunt references further teaches "wherein said data structure is a table, a list or a tree data structure" (See Hunt: Fig. 2, element 230 and Page 3, [0028] where message digests are stored in a table of a database repository).

Conclusion

6. The prior art made of record

- A. U.S. Patent Application 2003/0005306
- B. U.S. Patent Application 2004/0111608
- C. U.S. Patent 5,742,807

6.1 The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

D. U.S. Patent 4,807,182

E. U.S. Patent 4,741,028

F. U.S. Patent Application 2005/0187970

G. U.S. Patent Application 2002/0029347


H. U.S. Patent Application 2003/0188180

Contact Information

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuen S Lu whose telephone number is (571) 272-4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone are unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 886-217-9197 (toll-free).

Art Unit: 2167

Kuen S. Lu 

Patent Examiner, Art Unit 2167

August 10, 2006